

Analysis And Detection Of Network Security Failures In Banking Systems Using Machine Learning

Dr Krishna Samalla , Polumati Suvarna Pushpa, Kandula Naga Siva Ganesh

¹Professor Dept of ECE Sreenidhi Institute of Science & Technology, India

²Assistant Professor Dept of CSE Vignana Bharathi Institute of Technology

³Assistant Professor Dept of ECE Sreenidhi Institute of Science & Technology, India

Abstract

Introduction: Digital banking depends on interconnected networks that support internet and mobile banking, automated teller machines, payment gateways, application programming interfaces, and core banking services. This connectivity improves access and transaction speed, but it also creates opportunities for credential abuse, malware communication, denial-of-service activity, unauthorized access, and data exfiltration. This paper reformulates the supplied case study into an IEEE-style technical article and presents a machine-learning (ML) framework for identifying suspicious network behaviour in banking environments. Publicly reported figures reproduced in the source document provide context: the Reserve Bank of India (RBI) reported 23,953 bank-fraud cases involving ₹36,014 crore in 2024–2025 for frauds of ₹1 lakh and above, while card/internet fraud represented 56.5% of reported cases by area of operation [1]. The proposed workflow combines traffic and authentication telemetry, pre-processing, feature extraction, supervised classification, anomaly detection, risk scoring, and analyst-led response. Model-performance figures in the source material are illustrative rather than experimentally validated; accordingly, no empirical superiority claim is made. The paper also discusses evaluation measures, false alarms, operational limitations, and steps required for a reproducible implementation.

Keywords: Banking cyber security, network intrusion detection, machine learning, anomaly detection, fraud analytics, security monitoring..

Objectives: Detect known and previously unseen anomalies in near real time (ii) distinguish attacks from faults to route incidents correctly (iii) keep false alarms low enough for a Security Operations Centre (SOC) to act on and (iv) provide explainable output suitable for audit and regulatory review.

Methods: The framework has five layers, shown in Figure 1. Each layer has a single responsibility, which keeps the design modular and lets banks adopt it incrementally besides existing SIEM and IDS tools

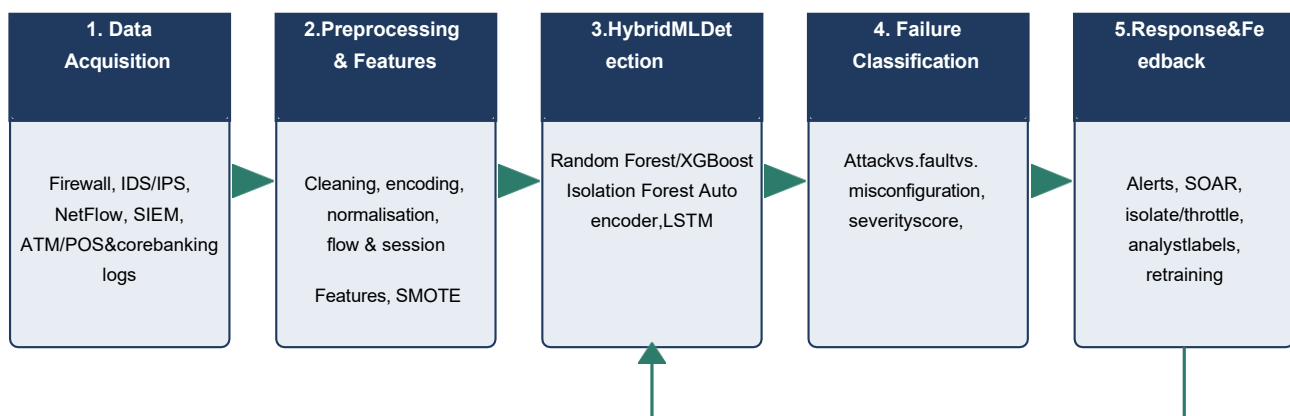


Figure 1. Layered security/fault detection framework with closed feedback loop

1. Introduction

Banks increasingly deliver services through connected digital channels, including web and mobile applications, ATMs, payment infrastructure, internal workstations, databases, and third-party interfaces. The resulting network activity is high-volume and time-sensitive. Security monitoring must therefore detect malicious behaviour while avoiding unnecessary interruptions to legitimate transactions.

The statistics cited in the supplied case study illustrate the importance of this problem. The RBI's Annual Report 2024–2025 records 23,953 fraud cases involving ₹36,014 crore for cases of ₹1 lakh and above. Of these, 13,516 were categorized as card/internet fraud, corresponding to 56.5% of reported cases by area of operation [1]. The same source indicates that reported card/internet cases fell from 29,082 in 2023–2024 to 13,516 in 2024–2025; a decrease in case count does not eliminate the need for continuous monitoring.

The source document also cites the Indian Computer Emergency Response Team (CERT-In) and sector partners' Digital Threat Report 2024, which describes phishing and compromised credentials as relevant risks to the banking, financial services, and insurance (BFSI) sector [3]. Separately, the National Payments Corporation of India (NPCI) statistics reproduced in the source report 24,508.96 million UPI transactions in August 2026 and 752 banks live on UPI [2]. These figures are included as supplied and should be checked against the original official releases before formal publication.

Rule-based controls, firewalls, signature-driven intrusion detection, access controls, and endpoint protection remain essential. ML can complement these controls by identifying combinations of signals that deviate from expected behavior. The objective of this paper is to describe a practical detection framework, a representative banking-network scenario, and an evaluation plan without presenting hypothetical model scores as measured evidence

2. Case-Study Scenario and Threat Model

Consider a bank network connecting core banking servers, internet and mobile banking platforms, ATMs, payment gateways, employee devices, databases, APIs, and external financial services. Under normal operating conditions, users authenticate, access permitted applications, and generate traffic patterns that vary by role, time, device, and service.

In the case scenario, an attacker obtains an employee's credentials and accesses a workstation. The workstation subsequently contacts an unfamiliar external host, generates repeated authentication failures, opens unexpected connections, and transfers more data outbound than its historical baseline. A single indicator may have a benign explanation; a correlated sequence can justify a higher-risk assessment. The system should classify events as normal, suspicious, or malicious and provide evidence for investigation rather than automatically treating every anomaly as a confirmed attack.

Class	Interpretation	Suggested response
Normal	Observed activity is consistent with the established baseline and policy.	Continue monitoring and retain telemetry.
Suspicious	One or more deviations require contextual review.	Create an alert and investigate supporting evidence.
Malicious	Multiple high-confidence indicators suggest compromise or an attack.	Escalate promptly and apply approved containment procedures.

3. Statistical Context

Table I summarizes the RBI figures reproduced in the supplied document. The data describe reported fraud cases and amounts involved not necessarily final net losses. The RBI source notes that reclassification and recoveries can affect interpretation [1].

Financial year	All fraud cases	Amount (₹ crore)	Card/internet cases	Card/internet amount (₹ crore)
2022–2023	13,494	18,981	6,699	278
2023–2024	36,060	12,230	29,082	1,457
2024–2025	23,953	36,014	13,516	520

Table I. Reported Bank Fraud And Card/Internet Fraud Figures Reproduced From The Source Document [1].

From 2023–2024 to 2024–2025, the reported total number of cases declined while the amount involved rose. The supplied document attributes part of the amount change to 122 cases amounting to ₹18,674 crore that were newly classified after re-examination [1]. These statistics should not be interpreted as a direct measure of cyberattack prevalence or as the expected loss for an individual bank.

The source also reports that advances accounted for 92.1% of the amount involved in 2024–2025, while card/internet fraud accounted for 56.5% of the number of cases [1]. This difference supports a risk model that considers both event frequency and potential impact. UPI activity figures from NPCI are relevant to operational scale, but transaction volume alone does not measure the number of network attacks [2].

4. Proposed ML-Based Detection Architecture

The proposed architecture is an analytical layer integrated with existing security controls. Its input streams may include network-flow records, firewall logs, authentication events, endpoint telemetry, server logs, and security-relevant payment-system events. Sensitive customer and account information should be minimized, access-controlled, and handled under the bank's privacy and retention policies

Stage	Processing activity	Output
1. Collection	Collect approved network, identity, endpoint, and server telemetry.	Time-aligned event records
2. Pre-processing	Validate fields, handle missing values, remove duplicates, normalize numerical variables, and label records where reliable labels exist.	Consistent analysis dataset
3. Feature extraction	Derive traffic rate, packet/byte counts, session	Feature vectors

	duration, connection frequency, port diversity, failed-login count, and destination novelty.	
4. Detection	Apply supervised classification to labeled patterns and anomaly detection to deviations from a baseline.	Class label and/or anomaly score
5. Risk scoring	Combine model outputs with asset criticality, identity context, and security policy.	Prioritized risk level
6. Alert and response	Present indicators, event timeline, and recommended actions to analysts.	Investigated alert and documented response

Table II. Proposed Detection workflow.

5. Network Features and Candidate Models

Candidate features should describe behaviour rather than rely on one isolated field. Examples include source and destination IP addresses, ports, protocol, packet count, transferred bytes, connection duration, failed authentication attempts, connection frequency, and short-term traffic rate. Additional context can include device identity, user role, time-of-day profile, and whether a destination has previously been observed. IP addresses and identifiers should be handled carefully because they can be sensitive and may be unstable or misleading when used without context

Security event	Representative indicator	Candidate ML approach
Unauthorized access	Login or access pattern inconsistent with user history	Supervised classification

Brute-force attempt	Repeated failed logins in a short interval	Classification and threshold rules
DDoS-like activity	Abrupt traffic or connection-rate increase	Anomaly detection
Malware communication	Unusual outbound destinations or periodic connections	Classification plus anomaly detection
Possible data exfiltration	Outbound transfer substantially above baseline	Anomaly detection and policy checks
Port scanning	Connections across many ports or hosts	Classification and behavioral rules
Insider misuse	Deviation from expected access patterns	User/entity behaviour analytics
Account takeover	Unusual device, location, or authentication sequence	Classification with identity context
Network intrusion	Unexpected session or packet behaviour	Classification and anomaly detection

Table III. Example Mapping Between Security Events, Indicators, and Candidate Methods.

Supervised models such as logistic regression, support vector machines, random forests, and neural networks can be trained when representative labelled data are available. Isolation Forest or other anomaly-detection methods can be explored when confirmed attack labels are sparse. Model selection must account for class imbalance, interpretability, inference latency, compute cost, and the consequences of false positives and false negatives.

6. Case-Study Scenario and Threat Model

Assume an employee usually accesses internal banking applications during normal business hours from a managed workstation. On a particular day, the system observes several failed logins, a successful login from an unfamiliar device, contact with a new external IP

address, an increase in connection frequency, and outbound traffic above the workstation's baseline. Rather than treating each event independently, the detector evaluates the temporal sequence and related context

Alert field	Illustrative value
Severity	High — pending analyst validation
Event	Abnormal network behaviour
Source	Internal banking workstation
Indicators	Repeated login failures; unfamiliar external communication; increased connection rate; elevated outbound traffic
Recommended action	Verify user and device, inspect related logs, and restrict communication if justified by approved response policy

Table IV. Example Alert Structure; Not a Live Security Event

The alert should include evidence, timestamps, related events, model confidence or anomaly score, and the reason for prioritization. High-confidence cases can be escalated quickly; ambiguous cases should enter an analyst review queue. Containment actions should follow established incident-response procedures and must consider the risk of interrupting legitimate banking services.

7. Evaluation Methodology

Experimental. They are therefore not reproduced here as evidence of model performance. A publishable evaluation requires a documented dataset, train/validation/test separation, label definitions, pre-processing details, parameter settings, and repeatable experiments.

Evaluation should use precision, recall, F1-score, false-positive rate, false-negative rate, and detection latency. Accuracy alone can be misleading when malicious events are rare relative to normal traffic. Data should be split in a way that prevents leakage between training and testing, for example by time

period or by entity where appropriate. Results should be reported with the test-set composition and, where feasible, uncertainty estimates.

A suitable experimental plan is to: (1) define the threat categories and labelling process; (2) establish a rule-based baseline; (3) train candidate supervised and anomaly-detection models; (4) tune thresholds on validation data; (5) evaluate on a held-out test set; (6) inspect false alarms and missed detections; and (7) test inference latency and resource use under realistic traffic volumes. Results from a controlled dataset should not be presented as proof of production effectiveness without operational validation.

8. False Positives, False Negatives, and Operational Controls

A false positive occurs when legitimate activity is flagged as malicious—for example, a planned migration or a legitimate increase in payment traffic. Excessive false positives can overwhelm analysts and disrupt customers. A false negative occurs when malicious activity is classified as normal, potentially allowing an intruder to remain active. Thresholds should be selected according to the impact of each error, and model outputs should be reviewed alongside deterministic controls and analyst judgment.

ML should not replace firewalls, multi-factor authentication, encryption, endpoint protection, intrusion detection/prevention, access governance, logging, or incident-response procedures. It should augment these controls by correlating signals and prioritizing investigations. Model drift, changes in banking applications, new attack behavior, data quality issues, and compute constraints require ongoing monitoring and periodic reassessment

9. Limitations and Future Work

The present work is a case-study framework based on the supplied document and public statistics cited therein. It does not report a newly conducted experiment, use a disclosed operational banking dataset, or establish measured detection performance. Public fraud aggregates also cannot directly establish the effectiveness of a network-level ML detector. The numerical values for UPI activity and threat reporting should be verified against the original releases before submission.

Future work should implement a reproducible test pipeline; assess streaming detection; investigate explainable AI methods; incorporate user/entity behaviour analytics evaluate graph-based

relationships among accounts, devices, IP addresses, and services; and integrate alerts with security information and event management (SIEM) and security operations centre workflows. Any automated containment should be limited to carefully validated, policy-governed cases with audit trails and rollback mechanisms

10. Conclusion

Banking networks must distinguish malicious behaviour from a large volume of legitimate activity across digital channels. The case-study statistics highlight the scale of reported banking fraud and digital payment activity, while also requiring careful interpretation of reported cases and amounts. This paper presents an ML-assisted detection workflow that combines telemetry collection, pre-processing, feature extraction, supervised classification, anomaly detection, risk scoring, and analyst-led response. The framework is intended to complement established security controls, not replace them. Its effectiveness remains a research question until it is demonstrated through a documented dataset, reproducible experiments, operational testing, and evaluation of false alarms, missed detections, and response latency

The proposed five-layer framework joins supervised, unsupervised and sequence-based learning with explainable classification and a feedback loop. It improves on rule-based monitoring by detecting novel threats, separating attacks from faults, and keeping humans in control of high-impact responses, giving banks a practical route to stronger and more resilient network security

References

- [1] Reserve Bank of India, Annual Report 2024–2025, Chapter VI, “Regulation, Supervision and Financial Stability,” Tables VI.2 and VI.3, Mumbai, India, 2025.
- [2] National Payments Corporation of India, “UPI Product Statistics,” monthly statistics for 2026–2027. [Online]. Available: <https://www.npci.org.in/what-we-do/upi/product-statistics>. Accessed: Oct. 9, 2026.
- [3] Indian Computer Emergency Response Team (CERT-In), SISA, and CSIRT-Fin, Digital Threat Report 2024 for the Banking, Financial Services and Insurance (BFSI) Sector, 2024..
- [4] Reserve Bank of India, Annual Report 2023–2024, Chapter VI, “Regulation, Supervision and Financial Stability,” discussion of fraud vintage and detection time lag, Mumbai, India, 2024.

- [5] Velitlaoreet id donecultricestinciduntarcu non sodalesneque. Non curabitur gravida arcu ac tortordignissim convallis aenean et.All
- [6] Breiman, L., "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [7] Bolton, R. J., and Hand, D. J., "Statistical Fraud Detection: A Review," Statistical Science, vol. 17, no. 3, pp. 235–255, 2002.
- [8] Fawcett, T., and Provost, F., "Adaptive Fraud Detection," Data Mining and Knowledge Discovery, vol. 1, pp. 291–316, 1997.
- [9] Phua, C., Lee, V., Smith, K., and Gayler, R., "A Comprehensive Survey of Data Mining-Based Fraud Detection Research," arXiv preprint arXiv:1009.6119, 2010.
- [10] Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., and Adams, N. M., "Transaction Aggregation as a Strategy for Credit Card Fraud Detection," Data Mining and Knowledge Discovery, vol. 18, pp. 30–55, 2009.
- [11] . Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., and Bontempi, G., "Learned Lessons in Credit Card Fraud Detection from a Practitioner Perspective," Expert Systems with Applications, vol. 41, no. 10, pp. 4915–4928, 2014.
- [12] Dal Pozzolo, A., Caelen, O., Johnson, R. A., and Bontempi, G., "Calibrating Probability with Undersampling for Unbalanced Classification," IEEE Symposium Series on Computational Intelligence, pp. 159–166, 2015. 8. Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., and Caelen, O., "Sequence Classification for Credit-Card Fraud Detection," Expert Systems with Applications, vol. 100, pp. 234–245, 2018.
- [13] Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., and Bontempi, G., "SCARFF: A Scalable Framework for Streaming Credit Card Fraud Detection with Spark," Information Fusion, vol. 41, pp. 182–194, 2018.
- [14] Carcillo, F., Le Borgne, Y.-A., Caelen, O., and Bontempi, G., "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection," Information Sciences, vol. 557, pp. 317–331, 2021.
- [15] . Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P., "SMOTE: Synthetic Minority Over-Sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
- [16] He, H., and Garcia, E. A., "Learning from Imbalanced Data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.
- [17] Saito, T., and Rehmsmeier, M., "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets,

Author Biography

Dr Krishna Samalla Ph D in Electronics & Communication and Ph D in Computer Science Engineering working as a Professor at Sreenidhi Institute of Science & Technology, He Published several papers in Reputed Journals and conferences and also completed Research Projects from DRDO and Central Government Agencies ,He is guiding many UG & PG Scholars in Engineering stream